Tool grafico per Vic-20

di Michele Morini - Monza (MI)

L'interprete V 2.0

Sia il Vic che il suo fratellino C64 sono dotati dell'interprete Basic V 2.0 della Microsoft che, senza avere grosse carenze, non si può certo definire pieno di fronzoli. Con l'uso del tool che vi propongo avremo a disposizione 51 comandi che apriranno nuovi orizzonti al nostro home. Per capire come questo sia possibile cominciamo con lo studiare la situazione, cioè l'interprete V 2.0.

Quanti di voi hanno seguito su MC gli articoli dedicati al Vic e al C64 sapranno ormai che ogni 1/60 di secondo il microprocessore pianta baracca e burattini e salta ad eseguire la routine di IRQ che, tra le altre cose, se abbiamo premuto un tasto provvede a sistemare il corrispondente codice ASCII nel buffer di tastiera (loc. 631-640). Il S.O. penserà poi a prelevare il carattere dalla coda del buffer e a visualizzarlo sul monitor o TV; questa situazione si protrae finché non premiamo il tasto di Return. A questo punto inizia la fase di input: la riga che abbiamo scritto su video viene trasferita in blocco nel buffer del Basic, cioè in una zona di Ram riservata all'interprete dove, con comodo, avviene il meccanismo di interpretazione (esecuzione) del comando. L'interprete, in questa fase, controlla i primi caratteri a sinistra della riga presente nel buffer: se riconosce un numero, questo viene interpretato come numero di linea di una riga di programma e tutti i caratteri che seguono sono sistemati in memoria al loro posto, altrimenti la linea viene considerata come un comando da eseguire in modo diretto. L'aspetto di cui ci interessiamo è il primo (per ora), cioè cosa avviene al momento di immagazzinare una riga facente parte di un programma Basic. Appena riconosciuta la linea, l'interprete passa il controllo alla routine di "tokenizzazione" che compie un'operazione fondamentale: cerca nella linea in esame le parole chiave del Basic e le sostituisce con dei codici di un byte detti appunto token. Questa codifica consente di ridurre drasticamente l'occupazione di Ram e di aumentare la velocità di esecuzione dei programmi. Ricavare il codice relativo ad uno statement è molto semplice: in Rom esiste una tabella dei comandi implementati, il token si ricava sommando 128 al numero d'ordine della parola in questa tavola; consideriamo, ad esempio, il comando PRINT: esso occupa il 25° posto e il suo token sarà dunque 153. Il Basic del Vic e C64 utilizza, per codificare i suoi statement, tutti i numeri da 128 a 203 e rimangono quindi disponibili, per le nostre nuove istruzioni, i codici da 204 a 254 (il 255 corrisponde al pi-greco). Dopo che la linea è stata compattata e messa in memoria, sorge il problema opposto, cioè durante il listing del programma, bisogna intercettare i token, cercare nella tabella la parola-chiave corrispondente e stamparla.

La routine di list deve anche risolvere alcune ambiguità che si possono presentare. Ad esempio, il codice 193, può essere sia il codice ASCII del carattere Shift + B, sia il token corrispondente al PEEK; per discriminare basta tener conto del carattere <"> e interpretare tutto quanto segue come codice ASCII fino alla chiusura delle virgolette o fino alla fine della riga. Finora abbiamo considerato il funzionamento del Basic in modo diretto (comandi da tastiera), dovremo ora, per concludere la nostra chiacchierata introduttiva,occuparci del suo funzionamento durante l'esecuzione del programma. Dopo il RUN viene caricata la prima linea nel buffer del Basic e ogni volta che, esaminando la riga, l'interprete trova un token effettua un salto alla routine che esegue il comando. A questo punto è d'uopo effettuare una distinzione negli statement: tra comandi e funzioni. La sostanziale differenza tra queste due categorie, un po' sommarie se vogliamo, sta nel fatto che una funzione compie un'operazione su di un valore numerico secondo una legge del tipo y = f(x), dove, dato un valore numerico in input (x), su cui è compiuta l'operazione f, viene restituito il risultato y. Una funzione, così come l'abbiamo definita, può essere lo statement A = SIN (B); viceversa, ed è questa la differenza che rende necessario un trattamento diversificato per i due tipi di istruzione, un comando non fornisce output numerico.

Non avrebbe infatti molto senso una linea del tipo IF (condizione) THENSIN (X), oppure, A = GOTO 100.

Per questi motivi esistono all'interno del Basic due routine distinte: una preposta all'esecuzione dei comandi e l'altra alla valutazione delle funzioni. Per consentire questa distinzione nell'esecuzione dei nuovi statement divideremo i token che li codificano, a loro volta, in due gruppi: tutti i codici compresi tra 204 e 244 li riserveremo ai comandi, i rimanenti, fino a 254 corrisponderanno alle nuove funzioni.

L'implementazione dei nuovi statement

Con quanto abbiamo visto finora dovrebbe essere chiaro che, per ampliare il normale Basic dei Commodore, è necessario rifare ex novo tutte le routine che, più o meno direttamente, entrano in contatto con i nuovi token: la routine di tokenizzazione, la routine di listing, quella che esegue i comandi e quella che valuta le funzioni. Per nostra fortuna mamma Commodore ha posto gli indirizzi di queste routine-chiave in una tabella in Ram; è quindi possibile comunicare al sistema la nuova posizione di questi programmi con estrema facilità. Per la gioia degli smanettoni ecco la tabella:

Locazione LB MB	Routine puntata
$0300-$0301	Routine d'errore
$0302-$0303	Comandi diretti
$0304-$0305	Routine di tokenizzazione
$0306-$0307	Routine di listing
$0308-$0309	Esegue i comandi
$030A-$030B	Valuta le funzioni

Sarà allora sufficiente modificare i vettori in tabella in modo che puntino alle nuove routine per renderle operative.

Per prima esaminiamo la routine di tokenizzazione: innanzitutto, per sistemare i vecchi statement, salteremo alla vecchia routine del Basic che provvederà a com-

Questo programma è disponibile su cassetta presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 125.

pattare i comandi originali. Successivamente confronteremo, carattere per carattere, la linea nel buffer con una lista delle nuove istruzioni posta in Ram. Ogni volta che riconosceremo uno statement basterà sommare 128 alla sua posizione nella tabella e otterremo il token corrispondente — una volta collocatolo al suo posto — e rimarrà da compattare la riga di tanti caratteri quanti ne abbiamo "guadagnati" con la sostituzione testé effettuata. Avanzeremo così fino ad incontrare uno 0, che segnala la fine della linea. Nel corso di queste operazioni ogni volta che si incontra il carattere <" > si esegue un loop fino a reperire o un altro <" > o uno 0; questo per evitare di tokenizzare parole-chiave che si trovano all'interno di stringhe da stampare. Infine vale la pena dare un'occhiata alla lista dei comandi cui fa riferimento questa routine. La tabella è formata dai codici ASCII delle nuove parole chiave, ma con una particolarità: al codice dell'ultimo carattere viene sommato 128. Questo permette di verificare la fine degli statement e quindi di contarli man mano che si procede. Di questo accorgimento bisogna tener conto se si vogliono introdurre altri comandi nel tool.

La routine di listing compie l'operazione opposta: nella scansione del listato Basic, quando trova un token, ricava il numero d'ordine dell'istruzione e ne stampa il nome che trova nella tabella. Anche qui bisogna tener presente che uno stesso codice può essere sia ASCII sia comando Basic. Per decidere controlleremo ogni ricorrenza del carattere <" > regolandoci come nella routine di tokenizzazione. Per finire, uno sguardo anche alle ultime due routine. Il loro funzionamento, a parte dei dettagli, è assolutamente analogo e possiamo tranquillamente fare un discorso valido per entrambe.

Con una chiamata della routine Charget del Basic ($0073) viene prelevato un carattere del programma e, se esso è nel range riservato ai nuovi comandi (204-244) o alle nuove funzioni (245-254), viene eseguito lo statement, altrimenti si esce con un salto alle routine del Basic Microsoft. La Microsoft ha adottato un sistema veramente geniale per effettuare la chiamata della routine che esegue il comando o la funzione: infatti, una volta intercettato il token e ricavatone il numero d'ordine, viene caricato nello stack del microprocessore l'indirizzo della routine da eseguire decrementato di uno e il programma termina con un JMP (salto). Così facendo, quando il 6502-6510 incontra un RTS (ritorna dalla subroutine) preleva l'indirizzo dove aveva effettuato un JSR (salto a subroutine) dallo STACK per riprendere l'esecuzione un byte più avanti.

Ora è chiaro che se noi abbiamo forzato nello STACK l'indirizzo del nostro programma -1, al primo RTS il processore ne inizierà l'esecuzione. Inutile aggiungere che in Ram è stata creata una tabella che contiene gli indirizzi dei nuovi comandi, opportunamente decrementati, nello stesso ordine con cui sono sistemati i nomi nella loro tabella.

Questa struttura consente di intervenire facilmente sul tool, una volta che si abbiano a disposizione le routine di list, di tokenizzazione e di esecuzione. Cambiare o sostituire dei comandi diventa veramente semplice a patto di avere un minimo di conoscenza di Assembler 6502 e un assemblatore. Il primo passo sarà cercare il nome da dare al nuovo comando, che NON deve contenere delle parole riservate all'interprete, ad esempio la parola PAINT contiene INT, già usato nel Basic V 2.0, potremo invece adoperare FILL senza creare conflitti. Dovremo poi sostituire, nella tabella contenente i nomi dei nuovi comandi il nome di quello nuovo ad uno di uguale lunghezza, tenendo presente la distinzione tra comandi e funzioni. Per rendere operativa la nuova routine, infine, ne collocheremo l'indirizzo decrementato di uno nella tabella degli indirizzi al posto del comando che abbiamo sostituito.

Ed ecco gli indirizzi utili:

Tabella word:	5F00-5FFF
Tabella indirizzi:	5A00-5A65
Routine di listing:	5E62-5A64
Routine di tokenizzazione:	5E00-5E61
Esegue comandi:	5E95-5EBA
Valuta funzioni:	5EBB-5EE3

Il tool

Questo programma fornisce un'esten-

```
 1 REM**********************
 2 REM*     TOOL           *
 3 REM*    GRAPHIC         *
 4 REM*    ORIENTED        *
 5 REM*PER VIC-20+16K*
 6 REM* VERSIONE 1.0 *
 7 REM*MICHELE MORINI*
 8 REM*  ##/5/''85    *
 9 REM**********************
 10 POKE56,80:SA=20480 CK=0
 30 FOR I=0TO4095
 40 OP=0:READOP$
 50 IFOP$="THEN100
 60 FORC=LEN(OP$)TO1STEP-1
 70 AS=ASC(MID$(OP$,C,1)):IFAS>64THENOP=OP+(AS-55)*16↑(LEN(OP$)-C):GOTO90
 80 OP=OP+VAL(MID$(OP$,C,1))*16↑(LEN(OP$)-C)
 90 NEXT
 100 POKESA+I,OP:CK=CK+OP:NEXT
 110 IFCK=516344THENPRINT"2000BUFFER ATTIVARE SYS20480":END
 120 PRINT"200ERRORE !!":PRINT"00RICONTROLLARE I DATA":PRINT"00BUON LAVORO !!":EN
 D
 1000 DATA20,E4,5E,A9,1E,8D,88,2,20,1B,E5,A2,,BD,30,50,20,D2,FF,E8,E0,30,D0,F5,A9
 ,50,85
 1001 DATA38,A9,20,85,2C,A9,,8D,,,20,20,18,E4,4C,74,C4,EA,EA,EA,EA,93,2A,45,58,
 54,45
 1002 DATA4E,44,45,44,20,42,41,53,49,43,20,56,31,2E,30,2A,11,20,42,59,20,4D,49,4B
 ,45,20
 1003 DATA53,4F,46,54,20,31,39,38,35,20,28,43,29,C,D,11,EA,EA,EA,20,8D,5B,20,7
 0,5B
 1004 DATAA2,3,B5,A8,95,A4,CA,10,F9,20,FD,CE,20,9E,D7,86,FB,86,A4,20,FD,CE,20,9E,
 D7,86
 1005 DATAFC,EA,EA,EA,A5,A8,C9,B0,B0,26,A5,A9,C9,A0,B0,20,4A,4A,4A,A,AA,BD,D0,
 5D,85
 1006 DATAFD,BD,D1,5D,85,FE,A5,A9,29,7,AA,BD,F8,5D,A4,A8,31,FD,F0,3,20,92,5D,E6,F
 0,E6
 1007 DATAA9,A5,AB,C5,A9,10,C6,A5,A4,85,FB,A5,A5,85,A9,E6,FC,E6,A8,A5,AA,C5,A8,10
 ,B4,60
 1008 DATAEA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA
 ,EA,EA
 1009 DATAEA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,20,F4,CE,20,F,DC,A9,E2,A0,E2,20,A2,
 DB,A5
 1010 DATA66,45,6E,85,6F,A5,61,20,2B,DA,20,F,DC,A9,1,A0,68,20,91,D3,A5,66,45,6E,8
 5,6F
 1011 DATAA5,61,20,12,DB,60,20,F4,CE,20,F,DC,A9,1,A0,68,20,91,D3,A9,66,45,6E,85,6
 F,A5
 1012 DATA61,20,2B,DA,20,F,DC,A9,E2,A0,E2,20,A2,DB,A5,66,45,6E,85,6F,A5,61,20,12,
 DB,60
 1013 DATA20,9E,CD,20,CA,DB,20,FD,CE,20,9E,CD,20,C7,DB,A9,57,A0,,20,8C,DA,A5,66,4
 5,6E
 1014 DATA85,6F,A5,61,20,12,DB,20,CC,DC,A9,5C,A0,,20,8C,DA,A5,66,45,6E,85,6F,A5,6
 1,20
 1015 DATA2B,DA,20,F,DC,A9,57,A0,,20,A2,DB,A5,66,45,6E,85,6F,20,53,D8,20,58,DC,4C
 ,F7,CE
 1016 DATA20,9E,D7,86,94,A9,1E,85,95,78,A9,51,8D,15,3,A9,B9,8D,14,3,58,60,78,C6,9
 5,D0
 1017 DATA2F,A9,1E,85,95,85,A6,A9,96,85,A4,A9,,,85,A3,85,A5,A8,B1,A3,29,F,C5,94,D0
 ,6,B1
 1018 DATAA5,49,80,91,A5,E6,A3,E6,A5,D0,EC,E6,A4,E6,A6,A5,A6,C9,20,D0,E2,4C,BF,EA
 ,A9,
 1019 DATA20,BD,FF,A9,1,AA,A8,20,BA,FF,A9,,20,BD,FF,A9,1,AA,A8,20,BA,FF,A9,,20,D5
 ,FF,60
 1020 DATA20,9E,CD,A0,2,B1,64,99,A4,,88,10,F8,20,82,D7,20,FD,CE,20,8A,CD,20,F7,D7
 ,A5,14
 1021 DATA85,A7,A5,15,85,A8,20,FD,CE,20,8A,CD,20,F7,D7,A5,14,85,A9,A5,15,85,AA,A5
 ,A4,A6
 1022 DATAA5,A4,A6,20,BD,FF,A9,1,AA,A8,20,BA,FF,A9,A7,A6,A9,A4,AA,20,D8,FF,60,20,
 8D,5B
 1023 DATAA2,3,B5,A8,95,B4,CA,10,F9,20,FD,CE,20,8A,CD,20,F7,D7,A5,14,85,A3,85,15,
 85,A4
 1024 DATA20,FD,CE,20,8A,CD,20,F7,D7,A5,14,85,F9,A5,15,85,FA,A2,,86,A7,20,F4,52,A
 9,FF
 1025 DATA85,A7,E6,A3,D0,2,E6,A4,A5,A4,C5,FA,D0,ED,A5,A3,C5,F9,90,E7,F0,E5,60,20,
 8D,5B
 1026 DATAA2,3,B5,A8,95,B4,CA,10,F9,E8,86,A7,86,A4,86,A3,20,F4,52,A9,FF,85,A7,18,
 A9,A
 1027 DATA65,A3,85,A3,A9,,65,A4,85,A4,C9,1,D0,E8,A5,A3,C9,72,90,E2,60,A5,A4,A4,A3
 ,20,91
 1028 DATAD3,20,E2,DA,20,F,DC,A9,2,A0,3D,20,91,D3,A5,61,20,12,DB,60,20,DA,52,20,6
 1,E2
 1029 DATA20,F,DC,A4,B5,A9,,20,91,D3,A5,61,20,2B,DA,20,49,D8,20,AA,D1,18,A5,B7,65
 ,65,85
 1030 DATAA9,48,A5,A5,85,AB,68,85,A5,20,DA,52,20,68,E2,20,F,DC,A4,B4,A9,,20,91,D3
```

sione di 51 nuove istruzioni, per la precisione 41 comandi e 10 funzioni, con una certa propensione per la grafica. La pagina grafica adottata per l'alta risoluzione ha le dimensioni di 160 × 176 pixel; non è il massimo che può fornire il Vic, ma è la combinazione che consente il miglior compromesso con un'accettabile occupazione della Ram disponibile. Quando è attivo il tool, la memoria del video è spostata a partire dalla locazioone 7680 e la Ram colore inizia a 38400 (come nel Vic inespanso) e la Ram riservata alla pagina grafica parte dalla locazione 4096 fino alla 7679. Questa disposizione delle "zone calde" del Vic consente di non alterare mai la pagina grafica ogni volta che pasticciamo lo schermo tornando in modo testo.

Prima di procedere all'esame dettagliato dei singoli statement è necessaria una precisazione: TUTTI i comandi che comportano il tracciamento in alta risoluzione sono dotati di un parametro che indica la modalità con cui avviene il disegno. Se il parametro assume valore 0 i pixel saranno spenti, se vale 1 saranno accesi e, se vale 2, saranno invertiti di stato; verranno cioè spenti i pixel trovati accesi e viceversa. Se è stato selezionato il modo Multicolor (vedremo come), che consente l'uso contemporaneo di 4 colori (1 di sfondo e 3 di tracciamento), questo parametro serve anche per selezionare il colore, ovvero: 0 spegne il pixel, 1 traccia col colore scelto per il cursore, 2 traccia col colore del bordo, 3 traccia usando il colore ausiliario e 4 inverte lo stato del pixel: un pixel acceso di colore 1 diventa di colore 2, il colore 3 viene spento e i pixel spenti vengono accesi di colore 3 (ausiliario). D'ora in poi indicheremo questo parametro con pt (che non vuol dire Poste e Telegrafi, ma plot-type).

I comandi

HGR: abilita l'alta risoluzione e NON cancella la pagina grafica. I colori usati in grafica sono quelli abilitati al momento di esecuzione del comando. Non richiede parametri.

CLS: cancella la pagina grafica.

TEXT: ritorna in modo testo cancellando il video. Lascia la pagina grafica inalterata.

COLOUR (SC, BO, IN, AUX): seleziona il colore dello schermo, del bordo, della traccia e il colore ausiliario. I parametri SC (schermo) e AUX (ausiliario) sono compresi tra 0 e 15; BO e IN sono limitati tra 0 e 7. Questa istruzione può essere usata anche con meno di quattro parametri, nel qual caso non vengono influenzati i colori omessi e viene automaticamente disabilitato il modo Multicolor. Per passare in Multicolor basta quindi usare questo comando con 4 parametri. Inoltre per cambiare i colori in alta risoluzione è necessario dare il comando prima di un HGR. Sono obbligatorie le parentesi e la virgola come separatore.

SOUND (X1, X2, X3, X4, VO): abilita gli oscillatori del Vic. I primi tre valori indicano la nota dei generatori sonori, X4 l'altezza del generatore di rumore e VO il volume. I primi quattro parametri devono essere compresi tra 128 e 254, VO tra 0 e 15. Anche qui il numero di parametri non è fisso, ma può andare da 1 a 5, sono obbligatorie le parentesi e la virgola come separatore.

PLOT pt, X, Y: accende, secondo le modalità indicate da pt, il punto di coordinate schermo X e Y. Provvede inoltre ad effettuare il clipping automatico e la gestione del modo Multicolor. I valori di X e Y devono essere compresi tra 0 e 255.

LINE pt, X1, Y1, X2, Y2: traccia una retta tra il punto di coordinate (X1, Y1) e il punto (X2, Y2). Tutte le variabili sono comprese tra 0 e 255 e separate da virgola.

FRAME pt, X1, Y1, X2, Y2: traccia il rettangolo avente come diagonale la congiungente i punti (X1, Y1) e (X2, Y2).

BOX pt, X1,Y1,X2,Y2: come FRAME, ma riempie il rettangolo.

AXIS pt, X, Y: traccia due assi ortogonali, uno verticale e uno orizzontale, di coordinate X e Y.

INVERSE: effettua il negativo di tutta la pagina grafica. Funziona anche in modo testo.

HCOL n, X, Y: modifica il colore di una zona di dimensioni 8 × 16 pixel entro cui cade il punto di coordinate X e Y. Il parametro n corrisponde al codice del colore da adottare entro la zona indicata. Come istruzione può apparire di dubbia utilità, ma con un po' di pazienza si possono ottenere degli effetti notevoli dividendo il video in zone in cui adoperare colori diversi.

CIRCLE pt, CX, CY, RX, RY: traccia una circonferenza o ellisse con centro nel punto (CX, CY) e raggio lungo l'asse delle X di lunghezza RX e, lungo l'asse delle Y, di lunghezza RY. Per ottenere una circonferenza il rapporto tra RY e RX deve essere circa 1.3.

ARC pt, CX, CY, RX, RY, A1, A2: traccia un arco di circonferenza o di ellisse con centro in (CX, CY) e raggi lungo gli assi X e Y rispettiva-

```
,A5,61                                                                      2
1031 DATA20,2B,DA,20,49,D8,20,AA,D1,18,A5,B6,65,65,85,A8,48,A5,A6,85,AA,68,85,A6
,24,A7
1032 DATA10,3,20,7,58,60,20,9E,D7,8A,48,20,FD,CE,20,9E,D7,8A,48,20,FD,CE,20,9E,D
7,8A
1033 DATA4A,4A,4A,4A,A,AA,BD,28,57,85,FD,BD,29,57,85,FE,18,A9,78,65,FE,85,FE,68,
4A,4A
1034 DATA4A,A8,E0,,F0,D,C6,FD,A5,FD,C9,FF,D0,2,C6,FE,CA,D0,F1,68,91,FD,60,20,9E,
D7,8A
1035 DATA4C,38,5D,60,20,9E,CD,A0,2,B1,64,99,A4,,88,10,F8,20,82,D7,A5,A4,F0,EB,A0
,,84
1036 DATA58,B1,A5,38,E9,30,B8,A8,A6,F9,A5,FA,88,30,1A,F0,1F,88,F0,24,88,F0,29,88
,F0,2D
1037 DATA88,F0,38,88,F0,44,88,F0,50,88,F0,5B,4C,48,D2,CA,30,55,E6,FB,50,F9,A8,88
,30,4D
1038 DATAE6,FC,50,F9,A8,88,30,45,C6,FC,50,F9,CA,30,3E,C6,FB,50,F9,CA,30,37,8A,48
,E6,FB
1039 DATA20,88,5D,68,AA,50,F2,A8,88,30,28,98,48,E6,FC,20,88,5D,68,A8,50,F2,A8,88
,30,19
1040 DATA98,48,C6,FC,20,88,5D,68,A8,50,F2,CA,30,B,8A,48,C6,FB,20,88,5D,68,AA,50,
F2,A4
1041 DATA58,C8,C4,A4,F0,3,4C,B3,53,60,20,8D,5B,A5,AA,85,FC,A5,AB,85,FB,A5,A8,85,
FA,A5
1042 DATAA9,85,F9,60,20,F4,CE,A0,2,B1,64,99,A4,,88,10,F8,20,82,D7,A2,,86,62,86,6
3,A4
1043 DATAA4,F0,45,C0,6,30,3,4C,48,D2,88,84,58,B1,A5,38,E9,30,F0,31,C9,8,10,EF,85
,59,A9
1044 DATA,85,5A,38,A5,A4,E5,58,A8;88,F0,F,6,59,26,5A,6,59,26,5A,6,59,26,5A,18,90
,EE,18
1045 DATAA5,59,65,63,85,63,A5,5A,65,62,85,62,A4,58,88,10,C3,4C,EC,58,20,F4,CE,A0
,2,B1
1046 DATA64,99,A4,,88,10,F8,20,82,D7,A2,,86,62,86,63,A4,A4,F0,3C,C0,11,30,3,4C,4
8,D2
1047 DATA88,84,58,B1,A5,38,E9,30,F0,28,C9,1,D0,EF,85,59,A9,,85,5A,38,A5,A4,E5,58
,A8,88
1048 DATAF0,7,6,59,26,5A,18,90,F6,A5,59,5,63,85,63,A5,5A,5,62,85,62,A4,58,88,10,
CC,4C
1049 DATAEC,58,20,9E,CD,A0,2,B1,64,99,A4,,88,10,F8,20,82,D7,4C,56,57,20,F4,CE,A0
,2,B1
1050 DATA64,99,A4,,88,10,F8,20,82,D7,A2,,86,62,86,63,A6,A4,F0,54,E0,5,30,3,4C,48
,D2,CA
1051 DATA8A,A8,B1,A5,C9,30,90,F4,C9,47,B0,F0,C9,3A,B0,8,38,E9,30,95,57,18,90,9,C
9,41
1052 DATA90,E0,38,E9,37,95,57,CA,10,DC,A6,A4,CA,B5,57,85,63,CA,30,1E,B5,57,A,A,A
,A,5
1053 DATA63,85,63,CA,30,11,B5,57,85,62,CA,30,A,B5,57,A,A,A,A,5,62,85,62,4C,EC,58
,20,9E
1054 DATAD7,86,FB,20,FD,CE,20,9E,D7,86,FC,A5,FC,C9,B0,B0,29,A5,FB,C9,A0,B0,23,4A
,4A,4A
1055 DATAA,AA,BD,D0,3D,85,FD,BD,D1,3D,85,FE,A5,FB,29,7,AA,BD,F8,3D,A4,FC,31,FD,F
0,2,A9
1056 DATAFF,A8,4C,91,D3,4C,48,D2,20,F3,56,A6,A8,86,58,A4,AB,88,20,12,57,98,48,B1
,59,48
1057 DATAB1,5B,48,C8,C0,16,90,1,88,68,91,5B,68,91,59,68,A8,88,C4,A9,10,E5,F0,E3,
E6,58
1058 DATAA5,AA,C5,58,B0,D5,A5,A8,85,58,A4,A9,20,12,57,A9,20,91,59,E6,58,A5,AA,C5
,58,B0
1059 DATAF1,60,20,F3,56,A6,A8,86,58,A4,A9,C8,20,12,57,98,48,B1,59,48,B1,5B,48,88
,10,1
1060 DATAC8,68,91,5B,68,91,59,68,A8,C8,C4,AB,90,E7,F0,E5,E6,58,A5,AA,C5,58,B0,D7
,A5,A8
1061 DATA85,58,A4,AB,20,12,57,A9,20,91,59,E6,58,A5,AA,C5,58,B0,F1,60,20,F3,56,A6
,AA,CA
1062 DATA86,58,A4,A9,20,12,57,B1,59,48,B1,5B,48,E6,58,A5,58,C9,17,B0,3,20,12,57,
68,91
1063 DATA5B,68,91,59,C6,58,C8,C4,AB,90,DF,F0,DD,C6,58,A5,58,30,4,C5,A8,B0,D1,A5,
A8,85
1064 DATA58,20,12,57,A4,A9,A9,20,91,59,C8,C4,AB,30,F9,F0,F7,60,20,F3,56,A6,A8,E8
,86,58
1065 DATAA4,A9,20,12,57,B1,59,48,B1,5B,48,C6,58,30,3,20,12,57,68,91,5B,68,91,59,
E6,58
1066 DATAC8,C4,AB,30,E3,F0,E1,E6,58,A5,AA,C5,58,10,D7,85,58,20,12,57,A4,A9,A9,20
,91,59
1067 DATAC8,C4,AB,30,F9,F0,F7,60,4C,48,D2,A9,4,85,BF,20,99,5B,20,70,5B,A9,15,C5,
AB,90
1068 DATAED,C5,A9,90,E9,A9,16,C5,AA,90,E3,C5,A8,90,DF,60,A5,58,A,AA,BD,28,57,85,
59,85
1069 DATA5B,BD,29,57,85,5A,18,69,78,85,5C,60,,1E,16,1E,2C,1E,42,1E,58,1E,6E,1E,8
4,1E
1070 DATA9A,1E,B0,1E,C6,1E,DC,1E,F2,1E,8,1F,1E,1F,34,1F,4A,1F,60,1F,76,1F,8C,1F,
A2,1F
```

(continua a pagina 118)

mente RX e RY, partendo da un angolo di ampiezza A1 fino all'angolo di ampiezza A2. I valori di CX, CY, RX, e RY sono compresi tra 0 e 255, A1 e A2 tra 0 e 65535.

COPY pt, X1, Y1, X2, Y2, X0, Y0: copia il rettangolo avente come diagonale la congiungente i punti (X1, Y1) e (X2, Y2) in un'altra zona della pagina grafica individuata da (X0, Y0), posizionandolo in modo che il vertice del rettangolo avente ascissa e ordinata minori coincida con (X0, Y0).

SCALE pt, X0, Y0, FX, FY: il funzionamento di questa istruzione è indissolubilmente legato alla DRAW (vedi); in pratica ha il compito di settare i parametri che influenzano lo statement DRAW. I valori di X0 e Y0 indicano le coordinate del punto da cui inizierà il tracciamento. FX e FY determinano, rispettivamente, il fattore di scala lungo l'asse X e Y (positivo e maggiore di uno).

DRAW "stringa": questo statement, molto versatile, esegue dei comandi di tracciamento contenuti in una stringa alfanumerica. I comandi eseguibili sono i seguenti:

'0' - Si sposta a destra di un pixel senza accenderlo.
'1' - Si sposta in basso di un pixel senza accenderlo.
'2' - Si sposta in alto di un pixel senza accenderlo.
'3' - Si sposta a sinistra di un pixel senza accenderlo.
'4' - Si sposta a destra di un pixel e lo accende.
'5' - Si sposta in basso di un pixel e lo accende.
'6' - Si sposta in alto di un pixel e lo accende.
'7' - Si sposta a sinistra di un pixel e lo accende.
'8' - Non opera.

Altri caratteri provocano errore.

È obbligatorio (badate, è una minaccia) eseguire almeno una istruzione SCALE prima di iniziare a tracciare, inoltre è possibile cambiare i parametri in ogni momento con un'altra SCALE.

Nel caso in cui un disegno non possa essere contenuto in un'unica stringa (255 comandi) è possibile usare più DRAW consecutivi.

HUE n: consente di cambiare il colore di tutta la pagina grafica. Non altera il colore della traccia.

CHAR "stringa", A, X0, Y0, FX, FY: questa istruzione consente di sovrapporre delle righe di testo in pagina grafica. Il tracciamento inizia dal punto (X0, Y0) e procede secondo due differenti fattori di scala lungo gli assi X e Y: rispettivamente FX e FY che DEVONO essere entrambi positivi e maggiori di uno. Infine il parametro numerico A seleziona il tipo di alfabeto che si vuole adottare per la stampa:

0 - Maiuscole.
1 - Shift + maiuscole.
2 - Maiuscole in reverse.
3 - Shift + maiuscole in reverse.
4 - Minuscole.
5 - Non utilizzato.
6 - Minuscole in campo inverso.

FLASH n: abilita il lampeggio, alla frequenza di circa 1 Hertz, e solo in modo testo, di tutte le scritte di colore di codice n (0-7).

FLASHOFF: spegne i caratteri lampeggianti.

UP X0, Y0, X1, Y1: effettua, in modo testo, lo scroll verso l'alto di una finestra rettangolare di diagonale (X0, Y0)-(X1, Y1). I parametri X0 e X1 (colonna) sono compresi tra 0 e 21, X1 e Y0 tra 0 e 22.

DOWN X0, Y0, X1, Y1: come UP, ma verso il basso.

LEFT X0, Y0, X1, Y1: come UP, ma verso sinistra.

RIGHT X0, Y0, X1, Y1: come UP, ma verso destra.

DISP (X, Y): è assolutamente analoga alla PRINT del Basic V 2.0 con la differenza che la stampa viene effettuata posizionando il cursore in X (colonna) e Y (riga).

IRQ N1, N2: abilita una routine di IRQ scritta dall'utente e posta all'indirizzo N1; il parametro N2 determina la frequenza di chiamata della routine che è inversamente proporzionale al valore del parametro. I valori di default sono 60095, per N1, e 18470 per N2.

RELEASE: ripristina l'IRQ di sistema ai valori di default.

PAUSE nn: effettua un ritardo proporzionale al valore di nn (0-65535). Un valore di 1000 corrisponde ad una pausa di poco più di un secondo.

DOKE N1, N2: effettua una POKE a 16 bit, mettendo nella locazione N1 la parte bassa di N2, e in N1 + 1 la parte alta.

CJUMP nn: equivale ad un GOTO calcolato dove nn può essere un'espressione qualsiasi che dia un risultato tra 0 e 65535.

CSUB nn: come CJUMP, ma esegue un GOSUB calcolato.

NOERR: ignora gli errori e passa allo statement successivo. Andateci piano! È un po' insidiosa...

SETERR nn: consente di gestire gli errori di sistema. Va usata sempre in testa al programma e, quando l'interprete trova un errore, passa il controllo alla linea nn. Potremo inoltre conoscere il codice dell'errore, che viene posto nella variabile riservata ST, e il numero di linea dove l'errore si è verificato, assegnato alla variabile LE% (da non usarsi nel programma).

ERR n: stampa il messaggio d'errore numero n. Unitamente alla SETERR, è utile per avvisare l'utente di eventuali errori senza fermare il programma (basta eseguire ERR ST).

```
(segue da pagina 117)                                                    3
1071 DATRB8:1F,CE,1F,E4,1F,20,FD,CE,20,8D,5B,A5,R4,F0,1C,A5,AB,85,FB,A5,AA,85,FC
,A9,40
1072 DATA5,92,85,62,A0,,84,A7,20,7F,57,A4,A7,C8,C4,R4,D0,F4,60,EA,EA,B1,A5,C9,40
,80,3
1073 DATR38:E9,40,A,A,26,62,85,61,A9,,85,57,A4,57,B1,61,85,5F,85,60,A5,A8,85,5
8,A9
1074 DATR8,85,59,A5,A9,85,5A,A9,,85,92,6,60,90,4,A9,1,85,92,20,88,5D,E6,FB,C6,5A
,D0,F7
1075 DATAC6,59,D0,E3,A5,A9,85,59,38,A5,FB,E9,8,85,FB,C6,59,D0,F5,E6,FC,C6,58:F0,
6,A5
1076 DATR5F,85,60,B0,C4,E6,57,A5,57,C9,8,D0,B0,A5,A8,85,58,38,A5,FC,E9,8,85,FC,C
6,58
1077 DATAD0,F5,A5,A9,85,59,18,A5,FB,69,8,85,FB,C6,59,D0,F5,66,62,60,20,8D,5B,A9,
,85,B0
1078 DATA85,B1,85,5A,85,5B,A2,C6,A0,E6,8C,B1,58,8C,B3,58,A5,A8,C5,AA,90,6,E5,AA,
85,59
1079 DATRB0,C,8E,B3,58:86,B0,38,A5,AA,E5,A8,85,59,A5,A9,C5,AB,90,6,E5,AB:85,58,B
0,C,8E
1080 DATRB1,58,86,B1,38,A5,AB,E5,A9,85,58,C5,59,90,11,A9,A9,8D,A0,58,8E,9E,58,24
,B1,30
1081 DATR1C,8C,9E,58,50,17,A9,A9,8D,9E,58,8E,A0,58,A5,58,A6,59,86,58,85,59,24,B0
,30,3
1082 DATA8C,A0,58,38,A5,58,E5,59,85,57,A5,AB,85,FB,A5,AA,85,FC,20,88,5D,A5,5A,F0
,4,24
1083 DATA5B:10,13,18,A5,59,65,5A,85,5A,A9,,65,5B:85,5B,A9,AB,C6,AA,50,11,38,A5,5
A,E5
1084 DATR57,85,5A,A5,5B,E9,,85,5B,E6,AB,C6,AA:C6,58,A5,58,C9,FF,D0,C1,60,A9,3,20
,FB,C3
1085 DATAA5,7B,48,A5,7A,48,A5,3A,48,A5,39,48,A9,8D,48,20,79,,20,B1,5A,4C,AE,C7,2
0,8A
1086 DATACD,20,F7,D7,A0,,B1,14,85,63,C8,B1,14:85,62,A2,,86,D,A2,90,A5,62:30,3,4C
,44,DC
1087 DATR4C,48,DC,20,8A,CD,20,F7,D7,A5,15,48,A5,14,48,20,FD,CE:20,8A,CD,20,F7,D7
,A5,14
1088 DATA85,B0,A5,15,85,B1,68,85,14,68,85,15,A0,,A5,B0,91,14,C8,A5,B1,91,14,60,2
0,9E
1089 DATAD7,E0,16,10,17,86,D3,20,FD,CE,20,9E,D7,E0,17,10,B:86,D6,20,F7,CE,20:87,
E5,4C
1090 DATA9D,CA,4C,48,D2,20,8A,CD,20,F7,D7,A5,14,85,F7,A5,15:85,F8,A9,65,8D,,3,A9
,59,8D
1091 DATA1,3,60,8A:85,90,A4,3A,C8,F0,3A,A9,CC:85,45,A9,C5,85,46,A0,,20,EB,D0,A0,
,A5,3A
1092 DATA91,47,C8,A5,39,91,47,A5,F7:85,14,A5,F8:85,15,A5,28,A6,2C,20,17,C6,90,11
,A5,5F
1093 DATRE9,1,85,7A,A5,60,E9,,85,7B:68,68,4C,EA,C7,R2,11,4C,3A,C4,20,9E,D7,E0,1F
,90,1
1094 DATA60,8A,A,AA,BD,26,C3,85,22,BD,27,C3,85,23,A0,,B1,22,48,29,7F,20,D2,FF,68
,30,E5
1095 DATRC8,D0,F2,20,8A,CD,20,F7,D7,A2,,CA,D0,FD,C6,14,D0,F9,A5,15:F0,4,C6,15,50
,F1,60
1096 DATA78,A9,BF,8D,14,3,A9,ER,8D,15,3,A9,26:8D,26,91,A9,48,8D,27,91,58,60,FE,5
C,43
1097 DATA5D,59,5D,9,5C,D0,5C,72,5D,3,5B:F1,5A,43,5B,B9,5A,AB,56,5F,56,19,56,D1,5
5,2A
1098 DATA59,7F,5A,E8,59,D2,5B,4F,53,94,53,FC,5E,A7,52,5F,50,59,52,E8,59,A2,51,D0
,59,FB
1099 DATA58,B0,5A,BD,58,68,5A,4B,59,AB,59,E3,5E,D,52,FB,51:73,5A,77,5A,3F,54,9C,
53,E
1100 DATR55,92,55,53,51,94,5C,83,5C,DA,58:21,55,85,54,53,54:25,51,F7,50,ER,ER,ER
,A9,95
1101 DATR8D,,3,A9,5E,8D,1,3,60,A9,64,D0,2,A9,70,78,8D:28,3,58,60,78,20,8A,CD,20,
F7,D7
1102 DATAA5,15,85,A9,A5,14,85,A8,20,FD,CE:20,8A,CD,20,F7,D7,A5:15,85,AB,A5,14,85
,AA,A2
1103 DATR1,B5,A8,9D,14,3,B5,AA,9D,26,91:CA,10,F3,58:60,20,8A,CD,20,F7,D7,4C,R3,C
8,20
1104 DATA9E,D7,86,92,20,FD,CE,20,9E,D7:86,FB,20,FD,CE,20,9E,D7,86,FC,A5,FB,48,A9
,,85
1105 DATAFB,20,88,5D,E6,FB,D0,F9,68,85,FB:85,B0,A9,,85,FC,20,88,5D,A5,B0:85,FB,E
6,FC
1106 DATRD0,F5,60,20,8D,5B,20,70,5B:A2,3,B5,A8,95,A4,CA,10,F9,85,AA,20,7,58,A5,A
5,85
1107 DATA9,85,AB,A5,A6,85,AA,A6,A4,E8:86,A8,20,7,58,A5,A7,85,A9,85,AB,A5,A6,85,
AA,A6
1108 DATAA4,E8,86,A8,20,7,58,A5,A6,85,A9,85,AA:A6,A5,E8:86,A9,A6,A7,F0,1,CA,86,A
B,20
1109 DATA7,58,60,EA,EA,EA,20,8D,5B,20,70,5B:E6,AA,E6,AB,A5,A8,85,FC,A5,A9,85,
FB,20
1110 DATA60,5D,E6,FB,A5,FB,C5,AB,90,F5,E6,FC,A5,FC,C5,AA:B0,6,A5,A9,85,FB,50,E7,
60,A5
1111 DATAA8,C5,AA,90,8,48,A5,AA,85,A8,60,85,AA:A5,A9,C5,AB,90,8,48,A5,AB,85,A9,6
```

BSV "nomefile", SA, EA: salva su nastro la zona di Ram a partire da SA (Start - Address) fino a EA-1 (End - Address) col nome voluto.

BLD: carica da nastro una parte di Ram precedentemente salvata con BSV. Non richiede parametri.

NOBREAK: disabilita il tasto di STOP, lo STOP + RESTORE e il comando di List.

BREAK: annulla l'effetto di NOBREAK.

INIT: rinfresca i puntatori alle routine del tool; è utile se si è pasticciato un po' troppo o per ripristinare la routine d'errore del Basic (dopo una NOERR o SETERR).

BASIC: esce dal tool effettuando un salto alla routine di inizializzazione del Basic.

Funzioni

PIXEL X, Y: effettua un test sul pixel di coordinate X e Y; se è acceso ritorna -1, altrimenti 0.

MOD (A, B): ritorna il resto divisione tra A e B, che possono essere due numeri reali qualsiasi.

JOY (0): effettua un test sul joystick e restituisce i seguenti valori: 1-alto, 2-destra, 4-basso, 8-sinistra e 128 se è premuto il tasto di fuoco. Il valore tra parentesi non ha nessuna influenza, ma è obbligatorio.

PDL (0) o PDL (1): legge una delle due paddle collegabili alla porta giochi del Vic e restituisce un valore tra 0 e 255. La selezione della paddle da testare è fatta tramite il valore tra parentesi che può essere o 0 o 1.

DEEK nn: effettua la PEEK a 16 bit alla locazione nn; è l'esatto complemento della DOKE.

HEX ("stringa"): effettua la conversione di una stringa esadecimale (max 4 caratteri) e restituisce un valore tra 0 e 65535.

BIN ("stringa"): effettua la conversione di una stringa binaria (max 16 caratteri) in intero decimale.

OCT ("stringa"): effettua la conversione di una stringa in ottale in intero decimale.

RAD (X): effettua la conversione in radianti del valore X dato in sessagesimali.

DEG (X): effettua la conversione in sessagesimali del valore di X dato in radianti.

P.S. in pagina grafica l'origine è posta in alto a sinistra, il verso positivo delle ascisse è da sinistra verso destra e quello delle ordinate dall'alto verso il basso. Le ascisse variano da 0 a 159 e le ordinate da 0 a 175.

Caricamento e uso

Dopo aver visto un lato della medaglia, quello migliore, passiamo all'altro, generalmente noto per essere meno gradevole. Si tratta di copiarsi i listati Basic contenenti le righe DATA dei codici esadecimali di 4096 byte: niente di simpatico. Vi conviene affrontare il lavoro in piccole dosi e con molta calma, magari dividendolo tra più persone. Il programma caricatore è dotato di checksum, ma, anche quando avete finito di copiare e il controllo è corretto, NON date la SYS di partenza e salvate il listato sorgente su nastro o floppy, questo per due buoni motivi: innanzitutto anche se il check-sum è esatto, se non è la vostra giornata fortunata, può darsi che due o più errori si compensino, in secondo luogo l'inizializzazione del tool distrugge il programma Basic in Ram. Successivamente, quando avremo controllato singolarmente i vari comandi, sarà inutile caricare ed eseguire il programma Basic, cosa che richiede molto tempo, e potremo salvare solo la Ram interessata con il comando, BSV "tool", 20480, 24576 (vers. 16K) per ricaricarlo eseguiremo LOAD "tool", 1,1 o semplicemente LOAD.

Per inizializzare il tool è necessario chiamare da Basic un'apposita routine con il comando SYS20480 (vers. 16K). A questo punto comparirà la scritta EXTENDED BASIC V1.0 e l'indicazione della memoria disponibile seguiti dal consueto e familiare READY. Sull'uso dei nuovi comandi non rimane molto da dire: sono assolutamente compatibili con il Basic V2.0 e non ci sono bug di sorta; quindi prima di sparare sul programmatore, controllate i vostri listati. L'unica cosa veramente importante riguarda l'uso dell'istruzione THEN che quando è seguita da uno statement del tool, richiede dopo di sé i ':' questo perché, dopo l'esecuzione del THEN, il controllo viene passato direttamente alla routine che esegue i comandi Microsoft, aggirando quella che noi abbiamo riscritto, provocando un SINTAX ERROR; inserendo i ':' l'inconveniente è superato. Infine una cosa di secondaria importanza: se usate i comandi in modo diretto noterete che l'istruzione viene eseguita correttamente ma, ogni tanto, compare un Sintax Error; non preoccupatevi, va bene così. MC

```
8,85                                                                      4
1112 DATAAB,60,20,9E,D7,86,92,A9,4,85,BF,20,FD,CE,20,9E,D7,A4,BF,96,A7,C6,BF,D0,
F2,60
1113 DATA46,FB,8,6,FB,A5,92,48,48,4A,85,92,20,92,5D,68,C9,4,D0,4,A9,2,D0,2,29,1,
85,92
1114 DATAE6,FB,20,92,5D,68,85,92,28,B0,2,C6,FB,60,EA,EA,EA,EA,AD,5,90,C9,FC,F0,1
6,A2
1115 DATA,BD,,1E,49,80,9D,,1E,BD,,1F,49,80,9D,,1F,E8,D0,ED,60,A9,10,85,FE,A9,,85
,FD,A8
1116 DATAA2,E,B1,FD,49,FF,91,FD,C8,D0,F7,E6,FE,CA,D0,F2,60,A2,,8E,,1,20,9E,D7,A8
,8A,AE
1117 DATA,1,9D,1,1,E8,E0,5,B0,B,20,73,,C0,29,F0,7,C0,2C,F0,E2,4C,8,CF,E0,3,F0,16
,A9,8
1118 DATA8D,86,2,AD,4,1,29,F,A,A,A,A,D,E,90,8D,E,90,90,10,A9,F7,2D,86,2,8D,86,2,
A9,F
1119 DATA2D,E,90,8D,E,90,AD,86,2,29,F8,8D,86,2,AD,3,1,29,7,D,86,2,8D,86,2,AD,2,1
,29,7
1120 DATA9,8,8D,F,90,AD,1,1,29,F,A,A,A,A,D,F,90,8D,F,90,60,20,9E,D7,E0,2,90,3,4C
,48,D2
1121 DATABD,8,90,A8,4C,A2,D3,20,9E,D7,78,A2,7F,8E,22,91,AC,20,91,CC,20,91,D0,F8,
A2,FF
1122 DATA8E,22,91,A2,F7,8E,20,91,58,AD,1F,91,CD,1F,91,D0,F8,48,29,1C,4A,C0,80,90
,2,9
1123 DATA10,A8,68,29,20,C9,20,98,6A,49,8F,A8,4C,A2,D3,A2,,8E,,1,20,9E,D7,A8,8A,A
E,,1
1124 DATA9D,1,1,E8,E0,6,B0,B,20,73,,C0,29,F0,7,C0,2C,F0,E2,4C,8,CF,CA,BD,1,1,9D,
A,90
1125 DATACA,10,F7,60,A2,,A9,1E,85,FE,86,FD,8A,A8,48,A2,14,91,FD,18,69,B,C8,D0,2,
E6,FE
1126 DATACA,D0,F3,68,18,69,1,C9,C,D0,E8,A9,17,8D,3,90,A9,94,8D,2,90,A9,FC,8D,5,9
0,A9
1127 DATAE,8D,,90,A0,,AD,86,2,99,,96,99,,97,C8,D0,F7,60,A9,10,A0,,A2,E,85,FE,84,
FD,98
1128 DATA91,FD,C8,D0,FB,E6,FE,CA,D0,F6,60,A9,F0,8D,5,90,A9,C,8D,,90,A9,96,8D,2,9
0,A9
1129 DATA2E,8D,3,90,A9,93,4C,D2,FF,20,9E,D7,86,92,20,FD,CE,20,9E,D7,86,FB,20,FD,
CE,20
1130 DATA9E,D7,86,FC,A9,8,2D,86,2,F0,3,4C,A5,5B,A5,FC,C9,B0,B0,2B,A5,FB,C9,A0,B0
,25,4A
1131 DATA4A,4A,A,AA,BD,D0,5D,85,FD,BD,D1,5D,85,FE,A5,FB,29,7,AA,BD,F8,5D,A4,FC,A
6,92
1132 DATAF0,9,E0,2,F0,C,11,FD,91,FD,60,49,FF,31,FD,91,FD,60,51,FD,91,FD,60,,10,B
0,10
1133 DATA60,11,10,12,C0,12,70,13,20,14,D0,14,80,15,30,16,E0,16,90,17,40,18,F0,18
,A0,19
1134 DATA50,1A,,1B,B0,1B,60,1C,10,1D,80,40,20,10,8,4,2,1,20,7C,C5,A0,5,B9,FB,1,F
0,57
1135 DATAC9,22,F0,47,C9,41,90,40,C9,5B,B0,3C,84,B1,A2,,86,B,38,FD,,5F,F0,13,C9,8
0,F0
1136 DATA16,BD,,5F,F0,27,30,3,E8,D0,F6,E6,B,A4,B1,A9,C8,B9,FB,1,E8,D0,E0,A6,B1,A
5,B,18
1137 DATA69,CC,9D,FB,1,C8,E8,B9,FB,1,9D,FB,1,D0,F6,A4,B1,C8,D0,B0,C8,B9,FB,1,F0,
6,C9
1138 DATA22,D0,F6,F0,F1,60,8,C9,FF,F0,2A,24,F,30,26,C9,CC,90,22,28,38,E9,CB,AA,8
4,49
1139 DATAA0,FF,CA,F0,8,C8,B9,,5F,10,FA,30,F5,C8,B9,,5F,30,5,20,D2,FF,D0,F5,4C,EF
,C6,28
1140 DATA4C,1A,C7,20,73,,C9,CC,90,19,C9,F5,B0,15,20,A6,5E,4C,AE,C7,E9,CB,A,A8,B9
,1,5A
1141 DATA48,B9,,5A,48,4C,73,,20,79,,4C,E7,C7,A9,,85,D,20,73,,C9,F5,90,13,C9,FF,B
0,F,E9
1142 DATACB,A,A8,B9,1,5A,48,B9,,5A,48,4C,73,,A5,7A,D0,2,C6,7B,C6,7A,4C,86,CE,A2,
B,BD
1143 DATAF0,5E,9D,,3,CA,10,F7,60,3A,C4,83,C4,,5E,62,5E,95,5E,BB,5E,48,6C,,C0,48,
47,D2
1144 DATA43,4C,D3,54,45,58,D4,43,4F,4C,4F,55,52,A8,53,4F,55,4E,44,A8,50,4C,4F,D4
,4C,49
1145 DATA4E,C5,46,52,41,4D,C5,42,4F,D8,41,58,49,D3,55,D0,44,4F,57,CE,4C,45,46,D4
,52,49
1146 DATA47,48,D4,44,49,53,50,A8,49,52,D1,52,45,4C,45,41,53,C5,49,4E,56,45,52,53
,C5,48
1147 DATA43,4F,CC,48,55,C5,42,41,53,49,C3,43,49,52,43,4C,C5,43,4F,50,D9,41,52,C3
,46,4C
1148 DATA41,53,48,4F,46,C6,46,4C,41,53,C8,50,41,55,53,C5,44,4F,4B,C5,43,4A,55,4D
,D0,43
1149 DATA53,55,C2,4E,4F,45,52,D2,53,45,54,45,52,D2,45,52,D2,49,4E,49,D4,42,53,D6
,42,4C
1150 DATAC4,4E,4F,42,52,45,41,CB,42,52,45,41,CB,53,43,41,4C,C5,44,52,41,D7,43,48
,41,D2
1151 DATA50,49,58,45,CC,4D,4F,44,A8,4A,4F,D9,50,44,CC,44,45,45,CB,48,45,58,A8,42
,49,4E
1152 DATAA8,4F,43,54,A8,44,45,47,A8,52,41,44,A8,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
```

HAI PERSO LA MEMORIA?

Ogni blackout o microinterruzione dell'energia elettrica provoca l'immediata cancellazione di tutti i dati inseriti nella memoria del tuo computer.
Qualche volta il danno rappresenta il lavoro di una intera giornata.
I GRUPPI DI CONTINUITÀ DIGITEK EVITANO QUESTI COSTOSISSIMI INCONVENIENTI.
Il gruppo di continuità DIGITEK ad onda trapezoidale stabilizzata unisce al costo contenuto eccellenti prestazioni e garantisce la totale eliminazione dei disturbi derivanti da fluttuazione, da instabilità, da interruzioni di energia elettrica.

NOVITÀ ASSOLUTA PER IBM PC - PC/XT

MOD. XT 700 COMPLETO DI PORTABATTERIE SU RUOTE PIVOTTANTI

La serie GCS "no stop" garantisce la totale eliminazione dei problemi di rete, (instabilità, microinterruzioni, disturbi di linea, black-out) su tutti i sistemi medio piccoli esistenti.

GCS 150	Potenza max	1°	convert.	150 W
GCS 300	"	1°	"	300 W
GCS 500	"	1°	"	500 W
GCS 600 e XT 700	"	1°	"	450 W
	"	2°	"	200 W
GCS 1000	"	1°	"	600 W
	"	2°	"	300 W
GCS 1300	"	1°	"	800 W
	"	2°	"	500 W
GCS 2000	"	1°	"	2400 W

GCS 2400 modulare a convertitori componibili da 400 a 1200 Wat cad. per un totale di 2400 W

Per richiedere Catalogo Generale, inviare L. 2000 in francobolli
GRUPPI C. MC
Cognome _____
Nome _____
Via _____
Città _____ Cap. _____

DIGITEK COMPUTER
VIA VALLI, 28 - 42011 BAGNOLO IN PIANO (Reggio Emilia)
Tel. (0522) 61623 r.a. - Telex 530156